Technical Note: Training Choices and Operating Regimes in Linear Associative Memory

Linear associative memory has a surprisingly large design space. Different training methods can produce substantially different rank, null-space structure, noise tolerance, cross-talk, and ability to complete incomplete inputs. These properties also change with storage load.

- **Hebbian / outer-product training**
  - Simple and local.
  - Works well when comfortably under capacity.
  - Redundancy can provide useful noise reduction and completion.
  - Cross-talk grows as storage load increases.
  - At and beyond capacity, interference between stored associations becomes increasingly important.
- **Pseudoinverse training**
  - Directly computes a least-squares association.
  - Can achieve exact recall when the associations are compatible with the available dimensions.
  - Tends to use the available input subspace efficiently.
  - Its behaviour becomes increasingly sensitive to conditioning as capacity is approached.
  - Provides a useful baseline for studying rank and observability.
- **Projection learning**
  - Constructs a mapping whose output lies in a learned subspace.
  - The projection is idempotent: applying it again leaves the result unchanged.
  - Naturally creates a null space.
  - The null space can suppress noise and irrelevant variation.
  - The same null space can destroy information needed for later observation or completion.
  - Particularly interesting when the stored patterns form a low-dimensional, redundant structure.
- **SGD-trained linear memory**
  - Ultimately approaches a least-squares solution under suitable conditions.
  - Training dynamics matter: different input directions are learned at different rates.
  - Early stopping acts as a form of spectral regularization, often suppressing weakly supported directions.
  - Consequently, effective rank and noise sensitivity can change substantially during training.
  - SGD also provides a convenient experimental way of approximating projection-like behaviour by training both input-to-output and output-to-output fixed-point associations.
- **Random-intermediate training**

    - Train input to a fixed random intermediate representation, then intermediate to output.
    - The two matrices can subsequently be multiplied into a single deployed linear mapping.
    - A small intermediate creates a deliberate bottleneck and can greatly increase the effective null space.
    - A large intermediate creates an overcomplete representation, potentially providing redundancy without necessarily losing input rank.
    - A structured random intermediate could be made into an explicit error-correcting code, with correction performed during recall.
    - The intermediate therefore provides an additional design dimension: compression versus redundancy.

Storage regime matters just as much as the training rule

- **Under capacity**
    - There is room for redundancy.
    - Noise reduction and completion can be strong.
    - A substantial null space may be beneficial if it primarily removes nuisance directions.
    - The memory need not use all available dimensions to perform useful association.
- **At capacity**
    - The memory is being pushed toward the limit of independent associations it can represent.
    - Conditioning becomes important.
    - Weight vectors can become large and recall becomes increasingly noise sensitive.
    - Small perturbations can produce disproportionately large recall errors.
    - Observability and robustness are increasingly in tension.
- **Over capacity**
    - Exact storage is no longer possible for arbitrary associations.
    - Recall can nevertheless remain **statistically significant** rather than simply becoming meaningless.
    - With sufficiently random training associations, the residual/cross-talk behaviour can approach a Gaussian-like distribution.
    - The individual training examples produce competing **push/pull effects** on the weight vectors.
    - These effects partially average one another as more associations are stored.
    - Consequently, the matrix norm can become **smaller than the large critical norm observed near capacity**, despite the additional stored associations.
    - Thus over-capacity recall can become weaker but statistically detectable, rather than suffering an abrupt transition from perfect recall to zero recall.

A useful research viewpoint

The question is therefore not simply:

> **Which linear associative-memory training method is best?**

It is:

> **Which combination of training method, storage load, intermediate dimensionality and desired invariance gives the best trade-off between recall, completion, noise rejection, cross-talk, rank and observability?**

This suggests a systematic benchmark should measure all of these properties rather than reporting recall accuracy alone.

For CCSLM, the particularly interesting possibility is that **context can divide a large associative-memory problem into multiple smaller memories**, keeping individual mappings under capacity where redundancy, completion and noise tolerance may be substantially better.